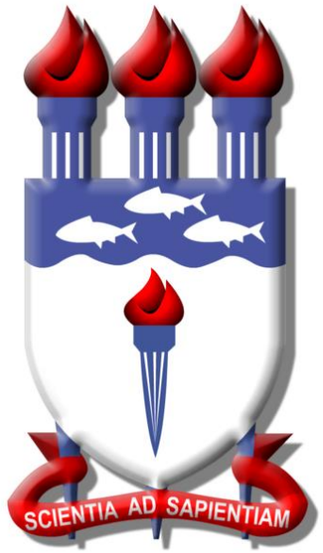


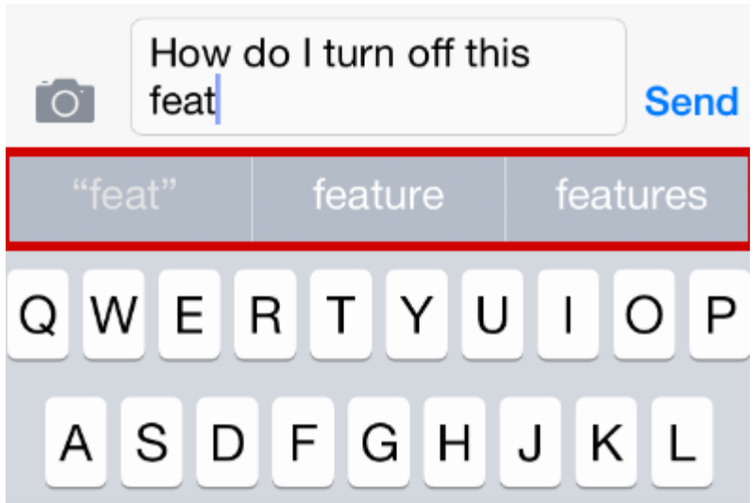
Estruturas de Dados / Programação 2

Trie

Equipe: Igor Hutson, Jhonnye Gabriel, Mateus Felismino, Hugo Melo

https://github.com/huffz/Huffman_Coding

Word Suggestion / Predictive Text


How do I turn off this feat
Send
"feat"
feature
features
Q W E R T Y U I O P
A S D F G H J K L


Vamos para o shop
shop
shopping
Shoptime
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123 Sym
Português

How can we do this?

Arrays?!

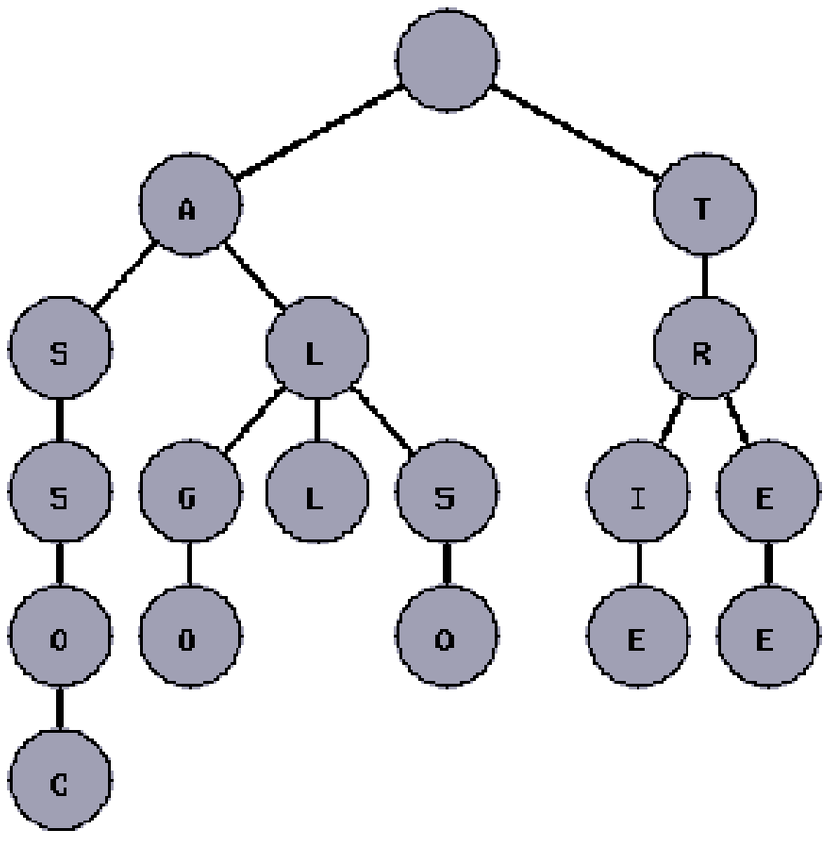
Hash Tables?!

Trie!

Trie

- Trie is an efficient information re***trie***val data structure.

- Used to store a dynamic set or associative key where the keys are usually strings.



Why should we use trie?

- Easier to Insert/search - O(n) - length of the string that you search or insert.
- No collisions.
- We can add each node according with new letters that are being added.

Definitions

- Leaf → A node which doesn’t have sons
- Internal Node → A node that completes a word in the middle of the TRIE.
- Edge → Connects two nodes.
- Path → Is a collection of nodes and edges.
- Depth → Is the length of the path from that node to the root.

Abstract Data Type:

Trie

Trie ADT

```
struct trie{
    int isWord;
    struct trie *suffix[MAX];
};
```

trie *create_node();

Int haveChildren(trie *current);

void insert(trie *head, char *str);

int search(trie *head, char *str);

trie *delete(trie *root, char *str, int depth);

Search( )

```
int search(trie *root, char *str){
        if(root == NULL)
        {
                return 0;
        }
        trie *current = root;

        while(*str){
                current = current->suffix[*str];
                if(current == NULL){
                        return 0;
                }
                str++;
        }
        return current->isWord;
}
```

Animação

ANIMAÇÃO_INTERATIVA_TRIE

Coming back to word suggestion

- We can predict which word the user is going to text by the prefix already texted.